Ethernet Traffic Measurements: A Case Study

José Luís Oliveira, Joaquim Arnaldo Martins, Joaquim Sousa Pinto

Resumo- **As redes locais de comunicação tem vindo a ser instaladas a um ritmo exponencial. Esta expansão, aliada ao crescimento de cada LAN, pela introdução de novos e mais exigentes sistemas, implica um aumento no tráfego de rede. O estudo destas variações é um dado importante, quer para a gestão, quer para o planeamento futuro da rede. Neste artigo são apresentadas algumas ferramentas que permitem estudar o tráfego numa rede Ethernet e algumas medidas de tráfego nomeadamente, tráfegos diários e semanais, impacto de novos sistemas e distribuições de tempos entre chegadas dos pacotes.**

Abstract- **Local Area Networks have been spread over the world during the last decade with an increasing installation rate. The expansion of each LAN, caused the introduction of new systems, usually implies more network traffic. It's important to the network manager to know the characteristics of this traffic in order to study, by simulation, the expected grow on data interchange and to plan future redesign of the network segments. This paper presents a set of tools to make network traffic analysis and to predict some resources requirements, over an Ethernet network. It presents as well some traffic measures, namely, daily utilization rate, impact of new hosts on network traffic and a study of the inter-packet time distribution.**

I. INTRODUÇÃO

The increasing number of installed Local Area Network puts new demands to the network manager when dealing with the control and the planning of the network configuration. Besides the crescent quantity of attached systems, there are a growing set of distributed services which have an important impact on the network traffic. Systems like UNIX, MSDOS PC, MAC, terminal servers, print servers, file servers, diskless workstations, remote printers and services such as NFS applications, E-mail, XWindows and distributed multimedia may be found in a nowadays network. All this panoply of systems and services have embedded different communication protocol's stacks (TCP/IP, OSI, AppleTalk, Netware, ...). The management of those types of networks has been mater of study and development during the past years either by leader vendors either by governmental standardization comities.

The ISO organization proposes a management scenario appropriate to the OSI layered architecture. In spite of being claimed more powerfully than other proposals, the standardization process is not complete which delays the development of practical solutions. The management information model is based on managed objects, abstract entities that represents data and communications resources. An object oriented design allows a single system to have several managed objects. These managed objects are able to report and notify information to another object acting as a manager, through a management protocol - the CMIP (Common Management Information Protocol).

The Internet community has proposed a set of RFCs (Request For Comments) that have been adopted by a crescent group of developers and users. The management information organization is based on a standard information model - the MIB (Management Information Base). The management protocol, SNMP - Simple Network Management Protocol, has already been included in the operating system of several hardware platforms. A second generation of the SNMP, the SNMPv2, is already defined and it incorporates some functionality's that allows some proximity with the OSI management protocol.

In both management architectures, Internet and ISO, the approach has been "start from the top", i.e., specify the models of protocols and system management architecture, namely the composition of the management information that is specific to each system.

Although the definition of these models is decisive to the beginning of development work, the network management task must "start from the bottom", i.e., it must perform some low level tasks in order to maintain a clean end-to-end connectivity. Examples are: monitor the communication medium, detect link anomalies and analyze the intensity of traffic.

After initial developments around the management models and on the systems management, the normalization groups begun some work that looks at first to the management of the network media. Examples of that work are the RMON MIB[1], the TR-RMON MIB[2], the BRIDGE MIB[3] and a few other ones that concentrate on the network and on the communication equipment monitoring [4]. The network traffic analyzes is crucial to plan the network short-term necessities and to evaluate the impact of new added equipment's.

While the Ethernet monitoring wasn't normalized we have developed some add-hoc tools, around a MSDOS PC environment, that help the characterization of the network activity [5,6]. This work was adapted to a RMON MIB agent which portable graphical user interface allows to construct a distributed monitoring scenario. In this paper we will describe briefly these developments, an analizer for packet interarrival time and some network measures coming from these tools.

II. MONITORING TOOLS

The first step towards an integrated monitoring architecture was the development of a Traffic Generator and a Traffic Analyzer, in order to evaluate how the network supports several load conditions and in order to characterize the network activity. This leads to a better understanding of the network behavior helping the manager to predict some areas of potential faults. The Traffic Generator allows to test the network or a specific machine behavior under different loads. The Traffic Analyzer reads information about all circulating packets like packet length, number of errors, number of broadcasts, protocol types, and the observed hosts' physical addresses. Another important task of the Analyzer is the capability to measure the packets' interarrival times. Since these tools are based on a common commercial PC, the number of simultaneous operations are limited to a few depending on the required processing power (Figure 1).

The Analyzer was adapted to an RMON Agent and consequently some others important capabilities have been integrated (like the Alarm and Event Report information) [7,8]. Besides the RMON compliance, the system still performs as an independent monitor since there are some monitoring tasks that aren't part of RMON MIB like the packet generation and the interarrival time measures (Figure 2).

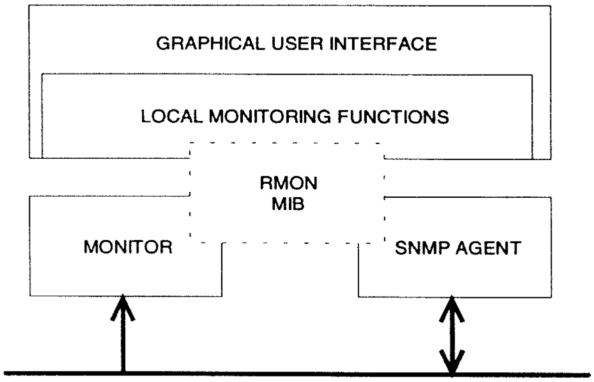

Figure 2 - The RMON Agent architecture

III. MEASURES IN AN ETHERNET SEGMENT

The University network is mainly composed by Ethernet LANs and there are several Class C TCP/IP networks (usually one for each department) that are interconnected through transparent bridges to the campus backbone. Due to the bridge forwarding method, the monitoring action performed by a RMON agent only applies in one side of the bridge considering this side of the topology as a single logical unbroken bus. The following considerations and measures were made in one of those network (inside the Electronics and Telecommunications Department). Besides the TCP/IP traffic there are also traffic coming from Novell and AppleTalk connections which allows to extend the total number of interconnected systems above the IP Class C capability. Nevertheless, the current quantity is near to 170 nodes distributed as follows: 4 Netware servers (with a total of 260 licenses), 30 UNIX workstations, 4 diskless workstations, 80 PC running Netware, TCP/IP or even a XWindows server, 40 Mac

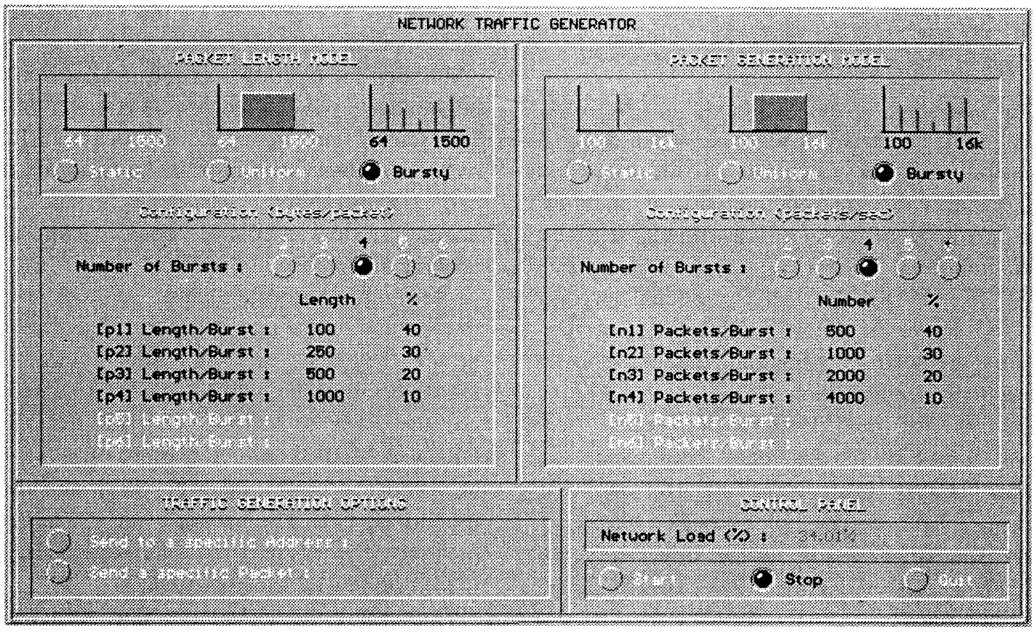

Figure 1 - The Traffic Generator user interface.

which use mainly Netware and 5 terminal servers. A reference must be made to some consuming services like NFS, to a VLSI classroom where 10 workstations and one server are running a centralized CAD application (at some discrete week intervals) and PC based classrooms which use the Netware infrastruture.

A. Traffic Measures

The measure of utilization rate on continuous mode allows to know how the network is used, what are the higher traffic intensity periods and their correspondent values. This information can be obtained either by the Traffic Analyser, either remotely from the RMON agent consulting its *Statistics* group objects.

The Figure 3, shows a weekday sampling, where it is emphasized the workday traffic if compared with the night lower traffic. The sampling was made over consecutive 30 minutes periods where can be detected 3 intervals with mean traffic above the 20%. The total mean was 6.13%, considering the overall period, and 9.12% if considering only the period among the 9 to 21 hours.

On Figure 4 it is plotted one selected workday measure over a 24 hours sampling and also the mean day computed along the week. We can see, for the mean day curve, that the afternoon is the more used network period with the peak load hour reached at 15 to 16:00, with a mean value of 13.91%.

B. Packets' Arrival Time

Measuring the packets' arrival time gives us two type of observations. The first, that we call *Packets' Interarrival Time*, consists in the measure of the time between two consecutive packet preambles. The other, the *Packets' InterGap Time*, measure the time space that separates two consecutive packets, since the end of the first one to the beginning of the second one (Figure 5).

The major problem on the measure of the packet generation time, with a software package, is the constraint for real-time operations. Considering the Ethernet transmission rate (10 Mbps) it is needed a time resolution of 0.1 μs that is a higher rate than a normal PC can handle. The Traffic Analyzer system can achieve a 20 μs resolution which is made by directly program the PC timer to generate 50,000 interrupts per second to construct a clock counter. This resolution represents the maximal absolute error for the packet time generation measurements.

However the statistic treatment that is associated with the resulting measures is typically based on histogram graphics which can attenuate partially this drift.

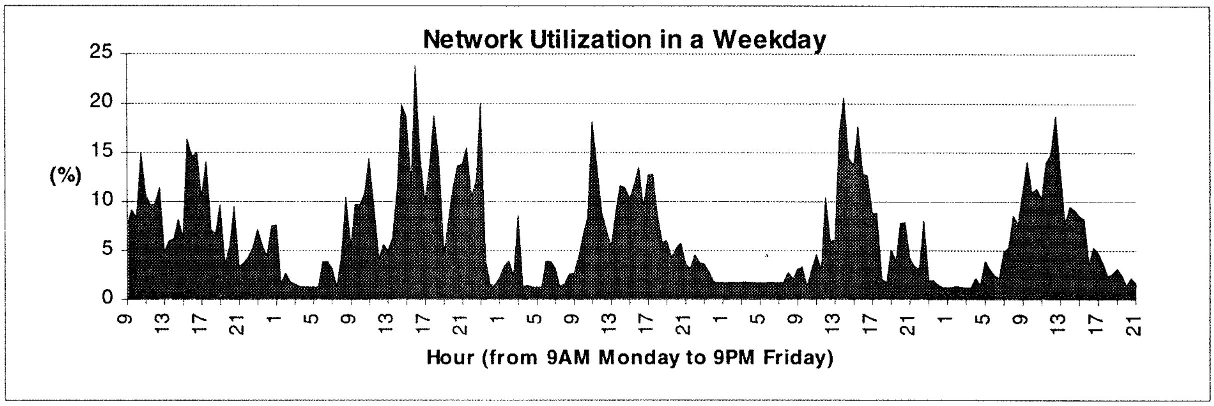

Figure 3- A Traffic sampling over a week period

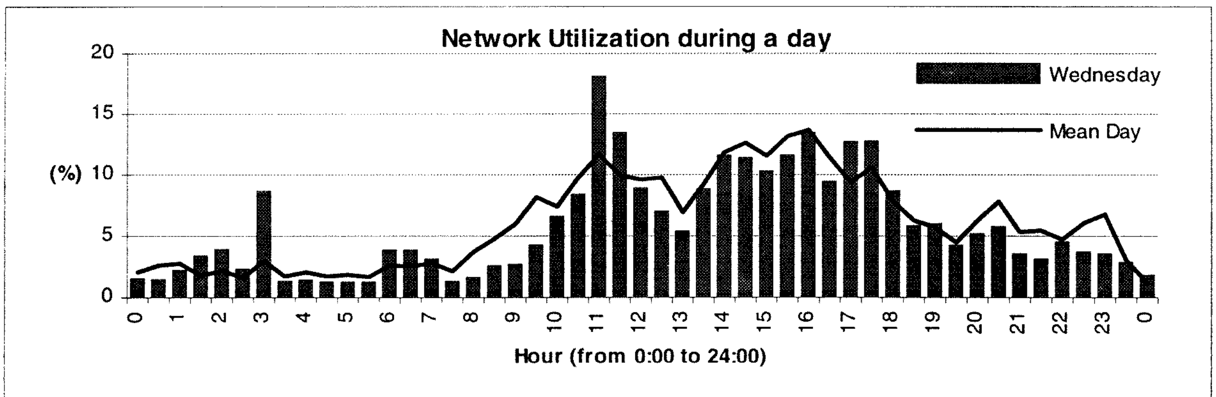

Figure 4 - A Traffic sampling over a 24 hour period

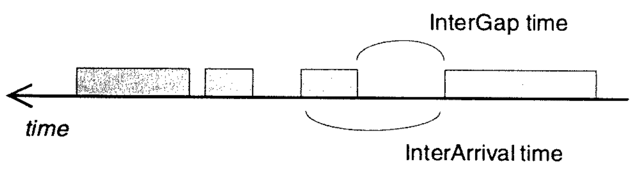

Figure 5 - Packets' InterArrival and InterGap Times

Nevertheless, the empirical statistic mean computed from these values tend to be a bit increased, due to the error associated with the measurement process.

Performing some measures during half a hour intervals we have arrived to the following results. The Interarrival time probability density function follows an unexpected curve that can be explained by the medium access method. When a packet transmission begins the medium remains unavailable until the end of the packet has been reached. After this, the network is free again for another transmission. So the delay between two consecutive headers depends on, and it is correlated to, the packet length. The overhead introduced by the packet transmission time can vary from a few μseconds, resulting from small packets (collisions) to 1220μs the time needed for a 1514 byte length packet. Since the packet length isn't uniformly distributed the interference on the Interarrival time distribution is affected with some peaks that are caused by the peaks of the packet length distribution (major peaks normally at 64, 1078 and 1500 bytes that implies delays respectively of 51, 862 and 1200 μs).

The Figure 6 shows a comparation between the measurement of interarrival time and intergap time empirical distribution. The main discrepancies, besides the spreading, is centered around 800 and 1200 μs.

Measuring the InterGap Times seems a good way to remove the packet length interference. As we will see some new particularities were found on this later method. The previous samplings were repeated using the same 30 minutes' interval but applying a Gap measurement. The Figure 7 represents the Probability Density Function (PDF) for these results. In spite of the new curve don't includes the old spikes the result is not so famous as it will be expected (a curve similar to the exponential probability density function). This drive us to other conclusions which are related with the way each service transfer information over the network. As the synchronous nature of a computer systems, the transmission mechanism also follows regular time intervals. For instance a file transfer operation is split over several packets which are send to the network in regular intervals, depending on the interface performance, and only affected by some other packets traveling between others systems and that momentarily monopolize the network media. It can be a bit ambitious to approximate the packet generation probability density function to an exponential one, since the packets, as we have discuss, are, most of the time, correlated. The no-correlation condition is fundamental when consequent events are supposed to follow this type of distribution [9].

On the graphic, it was plotted also an exponential curve (*Exp*) with the same mean as the empirical results one. The differences are caused by an higher density of the empirical PDF at higher time intervals. The curve *Exp2* shows a second exponential but for a mean that have an half value that the previous one. If the values above 3000 μs are removed, the curve fits better on the experimental PDF.

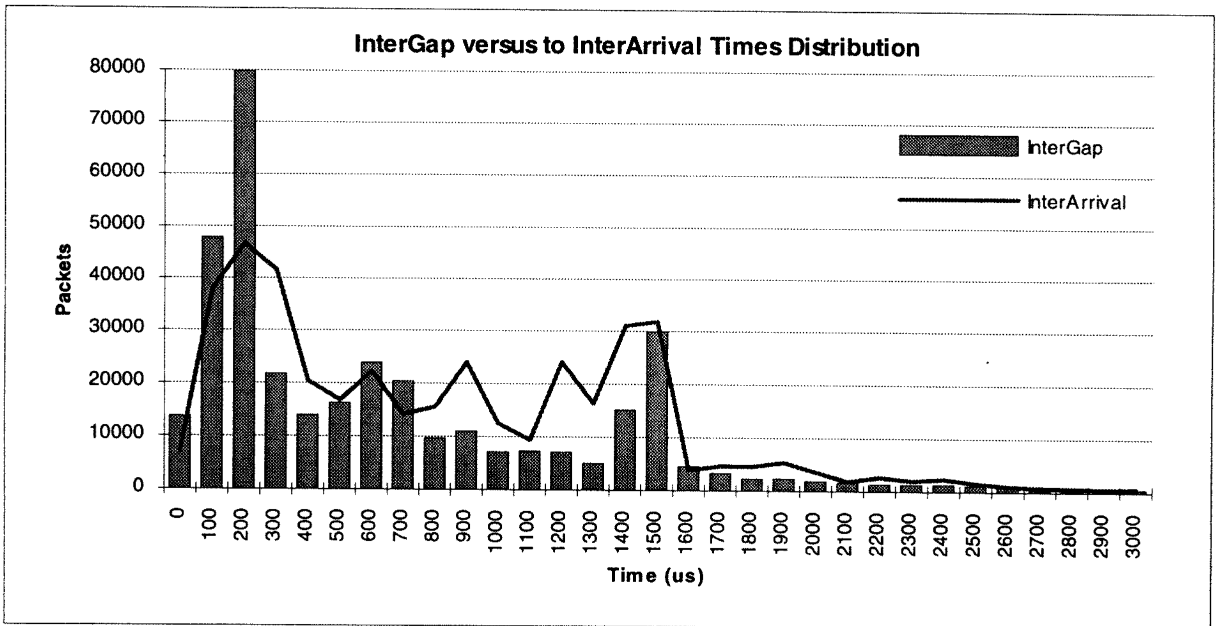

Figure 6 - Comparation between Empiricals PDF for packets interarrival and intergap times.

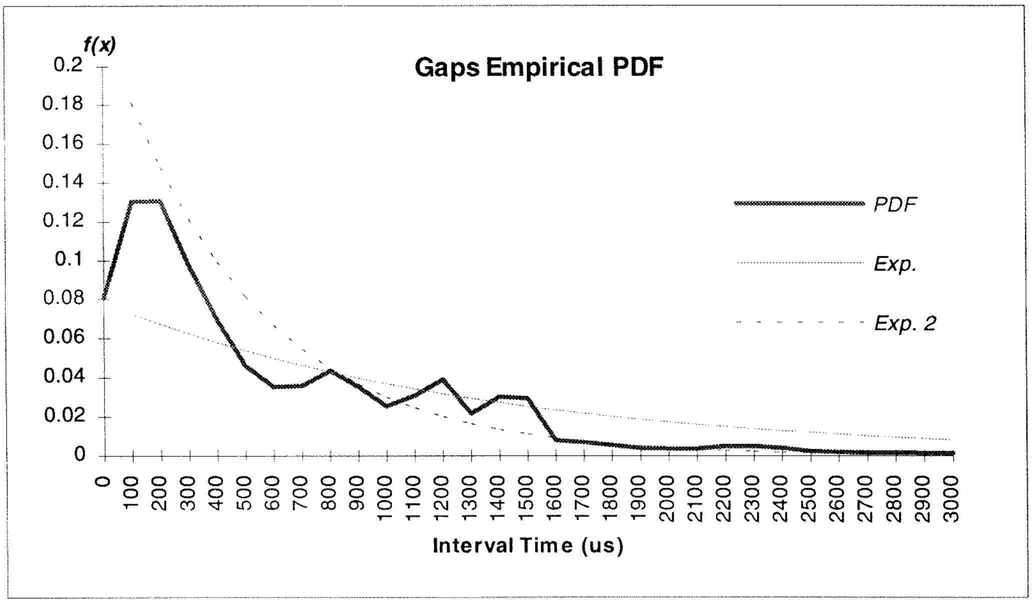

Figure 7 - Empirical Probability Density Function for packets' intergap times.

C. Hosts Study

The host addresses analyzer, study the number of hosts by each time interval and relates it with the measured traffic load. Other informations retrieved are the relations between packets and network load and also the influence of the network load on the errors ocurrence. On Figures 8 and 9 we can see a plot of these measures, made using 10 seconds sampling periods. While the packet number follows a curve near to linear, the error number increases quickly for values of load above the 40% showing a exponencial increasing rate.

The Figure 10 represents a hosts per traffic curve. It shows that for mean values, of active hosts, upper to 80, the network load is above the 50% of utilization rate.

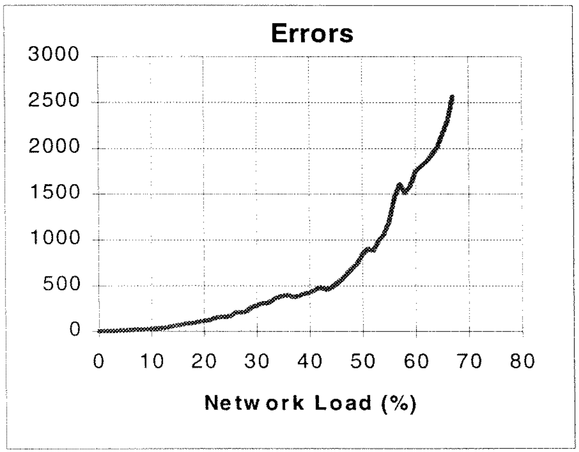

Figure 9 - Number of Errors by Network Load

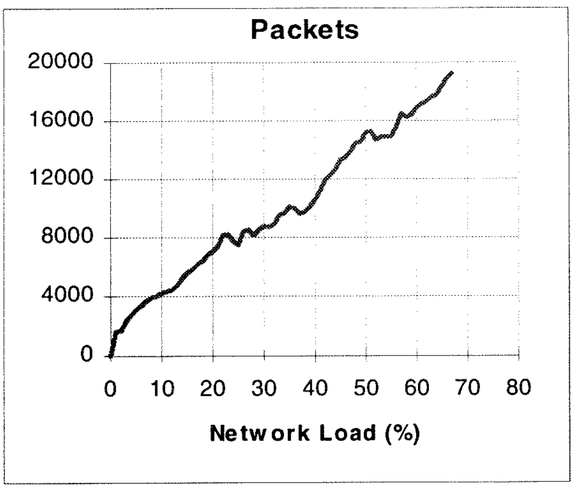

Figure 8 - Number of Packets by Network Load

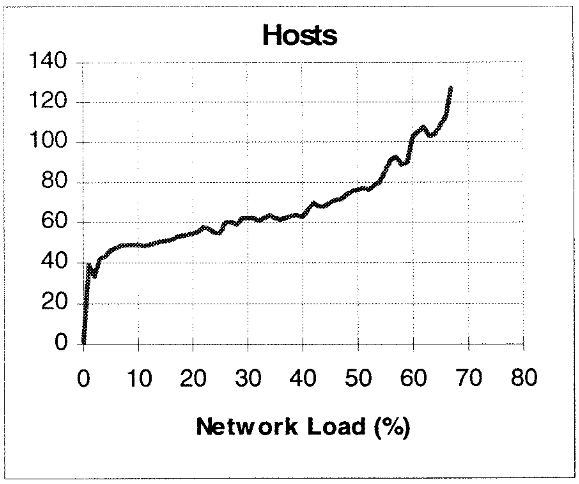

Figure 10 - Number of Hosts by Network Load

IV. CONCLUSIONS

Network Management must take as a priority the network monitoring to achieve a clean of errors media. Some standards have already been delivered on this field enabling the development of compactible products. We have present some of those tools, namely a Network Analyser that works either as a autonomous package either as a RMON agent. Some functions of this Analyser were explored like the measurement of the weekday traffic, interarrival and intergap times and hosts due to network load.

V. REFERENCES

[1] S. Waldbusser, "Remote Network Monitoring Management Information Base", *Internet Request for Comments 1271*, Carnegie Mellon University, Nov. 1991.

[2] S. Waldbusser, "Token Ring Extensions to the Remote Network Monitoring MIB", *Internet Request for Comments 1513*, Carnegie Mellon University, Sep. 1993.

[3] E. Decker, K. McCloghrie, P. Langille and A. Rijsinghani, "Definitions of Managed Objects for Source Routing Bridges", IAB, Sep. 1993

[4] F. Baker, D. Lostick and K. Tesink "Work Group Synopses", *The Simple Times*, November/December, 1993.

[5] J. Sousa Pinto, José L. Oliveira and J. Arnaldo Martins, "PC Workstation for Ethernet Monitoring and Management", *SBT/IEEE International Telecommunications Symposium*, Brazil, Sep. 1990.

[6] J. Arnaldo Martins, J. Sousa Pinto and José L. Oliveira, "Measuring, characterizing and generating LAN traffic", *IEEE ICCS'90 - International Conference on Communications Systems*, Singapore, Nov. 1990.

[7] Fernando Pipa, José L. Oliveira and J. Arnaldo Martins, "Desenvolvimento de um Agente de Monitoração para Redes TCP/IP", *Revista do DETUA*, Vol. 1, Nº 1, Janeiro 1994.

[8] Pedro Costa, Maurício Domingues, J. Arnaldo Martins and José L. Oliveira, "Monitorização e Visualização de Recursos num Ambiente Distribuído", *Revista do DETUA*, Vol. 1, Nº 1, Janeiro 1994.

[9] Averill Law and David Kelton, Simulation Modeling and Analysis, McGraw-Hill, 1982.